数式処理によるパラメトリック多項式最適化手法 (最適化手法の深化と広がり)

岩根, 秀直; 吉良, 知文; 穴井, 宏和

数理解析研究所講究録
第 1773 巻 2012 年 96-106

数式処理によるパラメトリック多項式最適化手法

岩根秀直
(株) 富士通研究所*
HIDENAO IWANE
FUJITSU LABORATORIES LTD

吉良知文
九州大学†
AKIFUMI KIRA
KYUSHU UNIVERSITY

穴井宏和
(株) 富士通研究所/九州大学‡
HIROKAZU ANAI
FUJITSU LABORATORIES LTD/KYUSHU UNIVERSITY

1 はじめに

多項式最適化問題とは、目的関数および制約条件が多項式の等式および不等式で記述された最適化問題のことである. 多項式最適化問題の記述能力は高く、工学および産業などでさまざまな応用がある. そのため半正定値緩和を用いるアプローチなど多項式最適化問題に対する研究が活発に行われているが, 非凸な問題やパラメトリックな最適化問題を正確に解くことは困難である.

数式処理を用いると変数や代数的数を記号計算により式を変形し正確な結果を得ることができる. 本稿では数式処理手法の一つである限量記号消去法を用いた多項式最適化問題に対する解法を紹介する.

2 限量記号消去法

数式処理は, 計算機上で代数的な記号演算を行い入力された式を式のまま変形し, 計算機代数とも呼ばれる. 多くの計算では浮動小数ではなく任意多倍長の整数または有理数を用い, 誤差のない結果を返す. 例えば多項式の最大公約因子や因数分解などの計算ができる. 数式処理を実現する数式処理システムは多数存在する. 例えば, 商用の数式処理システムとしては Maple や Mathematica, フリーで使用できるものとしては Risa/Asir [1)]などがある. 数式処理については [25] などを参照されたい.

本稿で扱う多項式最適化問題では, 目的関数や制約条件には多項式とその等式・不等式しか現れない. このように, 多項式の不等式で与えられるような制約条件の性質を調べたり, その制約条件下での多項式もしくは有理関数を最小にする問題は, **実代数幾何問題**と呼ばれ, 数式処理を用いた方法により正確に解くことができる.

---

*iwane@jp.fujitsu.com
†a-kira@math.kyushu-u.ac.jp
‡anai@jp.fujitsu.com

1) http://www.math.kobe-u.ac.jp/Asir/asir-ja.html

ここでは数式処理と区別するため浮動小数点を用いた従来の最適化手法を**数値手法**と呼ぶ. 数値手法は高速に計算ができるため活発に研究されているが, 数値手法だけを用いて非凸な問題やパラメトリック最適化問題・多目的最適化問題について十分な精度をもった結果を得ることは困難である.

数式処理のアルゴリズムのひとつである**限量記号消去** (Quantifier Elimination: QE) とは実数を動く変数に対して $\exists, \forall$ のような限量記号のついた一階述語論理式 (first-order formula) (多項式の等式, 不等式とそれらを $\wedge, \vee$ 等で結合した論理式) からそれと等価で限量記号のない論理式を計算するアルゴリズムのことである. 例えば $\exists x\ (x^2+bx+c=0)$ に対して, QE は x がない等価な論理式 $b^2-4c \geq 0$ を返す. この入力は 2 次関数 x^2+bx+c が x 軸と交わる条件を求める問題で, その判別式が非負であることに等価である.

QE は, 豊かな記述能力を持つ一階述語論理式によって表現される広い範囲の重要な応用問題を統一的にシステマチックに取り扱うことが可能であるため, 計算機科学や各種の理工学分野の研究者が QE を活用するようになっている.

ここで以下の章で必要な定義を行う.

定義 1
多項式の不等式または等式を原子論理式とし, 有限個の原子論理式を論理積または論理和の結合して表される集合のこと**半代数的集合** *(semi-algebraic set)* と定義する.

有限個の制約式からなる多項式最適化問題の制約条件を満たす点集合が半代数的集合で表現できることは明らかである.

定義 2
一階述語論理式のうち, 以下の形式で表される一階述語論理式を**冠頭標準型** *(prenex normal form)* と定義する.

$$\mathsf{Q}_{q+1}x_{q+1}\cdots\mathsf{Q}_r x_r\ (\psi(x_1,\ldots,x_r))$$

ここで, $\mathsf{Q}_i \in \{\exists, \forall\}$ で, ψ は限量記号がない論理式とする.

任意の一階述語論理式は, 冠頭標準型に変形することができる.

2.1 アルゴリズム

1930 年に A. Tarski [21] が実閉体 (real closed field) における決定手続きが存在することを証明し, QE アルゴリズムを示したが非常に効率の悪いものであった.

1975 年に George E. Collins により与えられた多項式系が符号が一定となるような領域に分割する **Cylindrical Algebraic Decomposition** (CAD) [6] を導入し, CAD による QE アルゴリズムが提案された. 現在でも CAD よりも効率的な QE アルゴリズムは発見されておらず, 不要な計算を回避するための方法 [3, 7, 18, 19] や数値手法と組み合わせた方法 [2, 8, 16, 20] など現在でも研究が続いている. また CAD の出力は符号が一定となる領域分割なので非常に有益で, QE 以外への応用も考えられる.

ところが, QE の計算量は最悪の場合, 限量記号がついた変数の数に対して二重指数的ということが示されている [9]. そのため応用上で重要な問題に関連した特定の制約条件に特化した専用アルゴリズムが研究されている.

そのひとつが限量記号がついた変数に関して低次（1 次・2 次）の多項式制約に対する **Virtual Substitution** (VS) [17] である. 他に, 多くの制御系設計の問題で現れる一変数多項式の正定性条件 (sign definite condition: SDC) に対する QE アルゴリズム [1] などがある.

QE に関する参考書としては [24] が良い. QE アルゴリズムの詳細やさまざまな応用例など紹介されている. 他には [5, 23] でアルゴリズムの詳細が確認できる.

2.2 QE ツール

QE はいくつかの数式処理システム上で実装されている. 以下, 代表的なものを簡単に紹介する.

QEPCAD [4]: 最も歴史ある CAD 専用の対話型コマンドで, SACLIB 上で動作する. 冠頭標準型の一階述語論理式のみを入力できる. QEPCAD はアルゴリズムの動作を変更する様々なスイッチが用意されている. さらに QE の出力だけでなく CAD の実行で得られるさまざまな情報も見ることができる. そのため CAD の学習にも用いることができる. また式の簡略化 (SLFQ) も実装されている. フリーでダウンロード[2]できる. 以下に実行例を示す.

```
=======================================================
                 Quantifier Elimination
                           in
             Elementary Algebra and Geometry
             Elementary Algebra and Geometry
                           by
       Partial Cylindrical Algebraic Decomposition

                Version B 1.58, 02 Mar 2011

                           by
                        Hoon Hong
                  (hhong@math.ncsu.edu)

With contributions by: Christopher W. Brown, George E.
Collins, Mark J. Encarnacion, Jeremy R. Johnson
Werner Krandick, Richard Liska, Scott McCallum,
Nicolas Robidoux, and Stanly Steinberg
=======================================================
Enter an informal description  between '[' and ']':
[example 1]
Enter a variable list:
(b,c,x)
Enter the number of free variables:
2
Enter a prenex formula:
(E x) [ x^2 + b x + c = 0].

=======================================================
Before Normalization >
finish

An equivalent quantifier-free formula:

4 c - b^2 <= 0
=====================  The End  =======================
```

REDLOG [3] [10]: フリーの数式処理システム REDUCE [4] 上の QE パッケージで, VS (rlqe) と CAD (rlcad) による QE コマンドや式の簡略化 (rlsimpl) が実装されている. 主に VS の実装に注力されているのが特徴である.

図 1 は REDLOG の実行例である. rlqe コマンドは VS を用いてできる限り変数を消去し, 次数が高く消去できない場合は限量記号のついた変数を残した論理式を復帰する. その場合は rlcad を適用する必要がある.

---

[2] http://www.cs.usna.edu/~qepcad/

[3] http://redlog.dolzmann.de/

[4] http://reduce-algebra.sourceforge.net/

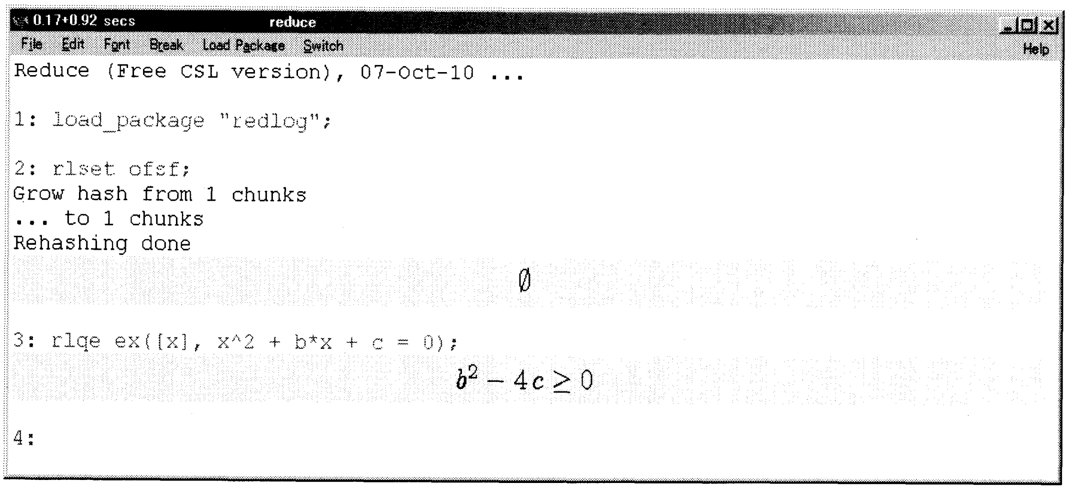

図 1: REDLOG 実行画面

Mathematica: 商用の数式処理システム Mathematica では CAD, VS による QE コマンド (Reduce, Resolve) が使用できる. QE コマンドでは InequalitySolvingOptions グループのシステムオプションを設定することで VS の使用等のスイッチを設定することができる. また QE の実行結果をそのまま描画コマンド (RegionPlot, RegionPlot3d) に渡して, 実行可能領域を描画することができる (図 2 参照). Mathematica の CAD の実装 [20] は, 数値手法と組み合わせて誤差なく高速化を実現した実装である.

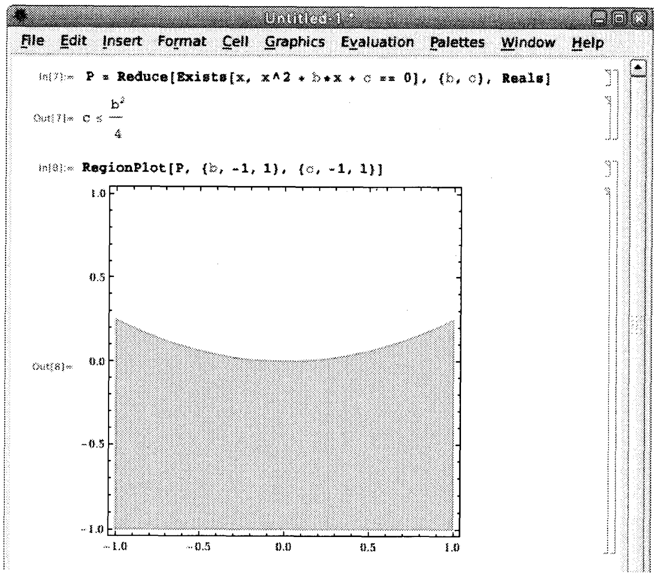

図 2: Mathematica 実行画面

SyNRAC: 現在筆者らの研究グループで商用 の Maple 上で QE パッケージ SyNRAC [14, 22] を開発中である. SyNRAC では CAD, VS, SDC が利用できる. また, 実行可能領域を描画する関数も利用可能である. 図 3 は SyNRAC の実行画面である. 現在, Maple ユーザ言語で実装した CAD による QE コマンドがフ

リーで使用できる. [5)]

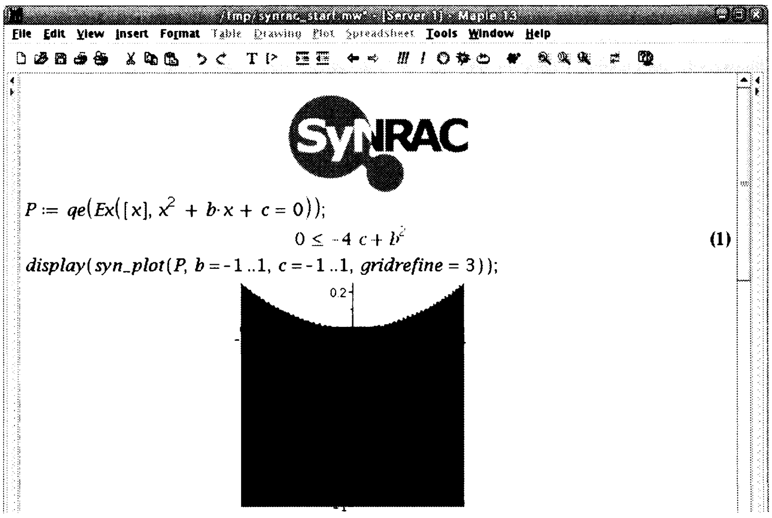

図 3: SyNRAC 実行画面

3 数式処理による多項式最適化手法

以下に示すパラメトリック多目的多項式最適化問題を考える.

$$\begin{array}{ll} \text{minimize} & (f_1(\boldsymbol{x},\boldsymbol{\theta}),\ldots,f_m(\boldsymbol{x},\boldsymbol{\theta})) \\ \text{subject to} & \varphi(\boldsymbol{x},\boldsymbol{\theta}) \end{array} \tag{1}$$

ここで, $\boldsymbol{x}=(x_1,\ldots,x_n)$ は**決定変数**, $\boldsymbol{\theta}=(\theta_1,\ldots,\theta_s)$ は**パラメータ**, $\varphi(\boldsymbol{x})$ は**半代数的集合**である. ここでは, 制約条件を満たす点集合はコンパクトであると仮定する.

多項式最適化問題を QE を用いて解くために, 新しい変数 $y_1,\ldots,y_m$ を導入して, 以下の一階述語論理式を考える.

$$\exists \boldsymbol{x}\ (\boldsymbol{y}\equiv \boldsymbol{f}(\boldsymbol{x},\boldsymbol{\theta})\ \wedge\ \varphi(\boldsymbol{x},\boldsymbol{\theta})) \tag{2}$$

ここで, $\boldsymbol{y}\equiv \boldsymbol{f}(\boldsymbol{x},\boldsymbol{\theta})$ は

$$y_1=f_1(\boldsymbol{x},\boldsymbol{\theta})\ \wedge\ \cdots\ \wedge\ y_m=f_m(\boldsymbol{x},\boldsymbol{\theta})$$

を表している. この一階述語論理式は, $y_1,\ldots,y_m$ が目的関数の実行可能領域に属する場合に真となる論理式である. したがって, この論理式から決定変数 $\boldsymbol{x}$ を消去するとパラメータ $\boldsymbol{\theta}$ を用いた目的関数の実行可能領域を表す論理式 $\psi_{Feasible}(\boldsymbol{\theta},\boldsymbol{y})$ を得ることができる.

$\psi_{Feasible}$ を用いて, パレート・フロントは次のように定式化できる.

$$\psi_{Feasible}(\boldsymbol{\theta},\boldsymbol{y})\wedge\neg\exists \boldsymbol{z}\ (\psi_{Feasible}(\boldsymbol{\theta},\boldsymbol{z})\wedge \boldsymbol{z}\le \boldsymbol{y}) \tag{3}$$

ここで, $\neg$ は否定を表す論理記号, $\boldsymbol{z}\le\boldsymbol{y}$ はすべての $i=1,\ldots,m$ について $z_i\le y_i$ かつ, 少なくとも一つの i で $z_i<y_i$ となることを示している. $\neg$ 以下の論理式で優越解が存在しないことを表現している. 式

[5)] http://jp.fujitsu.com/group/labs/techinfo/freeware/synrac/

(3) は以下の論理式と等価である. QEPCAD などの QE ツールは冠頭標準型を入力として受け付けるため以下のように変形する必要がある.

$$
\begin{aligned}
& \forall \boldsymbol{z}\ (\psi_{Feasible}(\boldsymbol{\theta}, \boldsymbol{y}) \wedge (\psi_{Feasible}(\boldsymbol{\theta}, \boldsymbol{z}) \rightarrow \neg(\boldsymbol{z} \leq \boldsymbol{y}))) \\
= \quad & \forall \boldsymbol{z}\ (\psi_{Feasible}(\boldsymbol{\theta}, \boldsymbol{y}) \wedge (\neg\psi_{Feasible}(\boldsymbol{\theta}, \boldsymbol{z}) \vee \neg(\boldsymbol{z} \leq \boldsymbol{y})))
\end{aligned}
$$

この論理式から, 変数 $\boldsymbol{z}$ を消去することでパレート・フロントを表す論理式 $\psi_{Pareto}(\boldsymbol{\theta}, \boldsymbol{y})$ を得ることができる.

最後に, パレート最適解を求める. パレート最適解は, その $\boldsymbol{f}$ による像がパレート・フロントとなるような実行可能解なので, 以下のように定式化できる.

$$
\exists \boldsymbol{y}\ (\boldsymbol{y} \equiv \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\theta}) \ \wedge\ \varphi(\boldsymbol{x}, \boldsymbol{\theta}) \ \wedge \psi_{Pareto}(\boldsymbol{\theta}, \boldsymbol{y}))
$$

この論理式から変数 $\boldsymbol{y}$ を消去すると, パレート最適解を表す論理式を得ることができる.

QE を用いると, 上記のように多項式最適化問題のすべての大域的最適解を誤差なく正確に求めることができる. QE は式の等価変換を行うだけで, その入力に次数や制約式の数などの制限はなく, 数値手法が得意でない非凸な最適化問題でも正確に解ける. また数式処理にとって, パラメータと変数の区別はないため多目的最適化問題でもパラメトリック最適化問題でも正確に解けることが特徴である.

ここでは QE 問題を 2 回解くことでパレート・フロントを求めているが, 以下の一階述語論理式で $\psi_{Feasible}$ を用いずに表すこともできる.

$$
\exists \boldsymbol{x}\ (\boldsymbol{y} \equiv \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\theta}) \wedge \varphi(\boldsymbol{x}, \boldsymbol{\theta}) \wedge \neg\exists \boldsymbol{u}\ (\varphi(\boldsymbol{u}, \boldsymbol{\theta}) \wedge \boldsymbol{f}(\boldsymbol{u}, \boldsymbol{\theta}) \leq \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\theta})))
$$

この論理式から $\boldsymbol{x}$ および $\boldsymbol{u}$ を消去することでパレート・フロントを直接求められる. しかし, 多くの QE アルゴリズムの計算量は変数の数に依存するため, 一般に (2) と (3) と 2 回に分けて計算したほうが効率的である. また, 実際に多目的最適化問題を解く場合には, パレート・フロントよりも目的関数の実行可能領域を得るほうが多くの情報を得ることができる.

また, 前述のように QEPCAD は構築した CAD の情報を見ることができ, 使用するツールによっては (2) を解く仮定で実行可能解を得ることも可能なので, 目的関数の実行可能領域さえ計算できれば十分な場合もある.

以下では単目的最適化, パラメトリック最適化, 多目的最適化の例を示す.

例 1

以下の単目的最適化問題を考える.

$$
\begin{array}{ll}
\text{minimize} & -x_1 - x_2 \\
\text{subjectto} & x_1 \geq 0 \ \wedge\ x_2 \geq 0 \ \wedge\ x_1^2 + x_2^2 \leq 1
\end{array}
$$

目的関数の実行可能領域は以下のように定式化される.

$$
\exists x_1 \exists x_2\ (y = -x_1 - x_2 \ \wedge\ x_1 \geq 0 \ \wedge\ x_2 \geq 0 \ \wedge\ x_1^2 + x_2^2 \leq 1)
$$

QE を用いて決定変数 x_1, x_2 を消去すると, 以下のような論理式を得る.

$$
y^2 \leq 2 \wedge y \leq 0
$$

最適値を表す論理式は以下から得られる.

$$
y^2 \leq 2 \wedge y \leq 0 \wedge \neg\exists z\ ((z^2 \leq 2 \wedge z \leq 0) \wedge z < y)
$$

QE を用いて変数 z を消去すると, 最適値を表す以下の論理式が得られる.

$$y^2 = 2 \wedge y \leq 0$$

つまり, 与えられた問題の最小値が $-\sqrt{2}$ であることが得られる. この結果を利用すると, 最適解は以下の一解述語論理式で表される.

$$\exists y \ (y = -x_1 - x_2 \ \wedge \ x_1 \geq 0 \ \wedge \ x_2 \geq 0 \ \wedge \ x_1^2 + x_2^2 \leq 1 \wedge y^2 = 2 \wedge y \leq 0)$$

ここから

$$x_1^2 + x_2^2 \leq 1 \wedge x_1 \geq 0 \wedge x_1^2 + 2x_1x_2 + x_2^2 \geq 2$$

最適解を表す論理式が得られる. 少し論理式が複雑なので, 最適解となる x_1 を以下の一解述語論理式を用いて求める.

$$\exists x_2 \ (x_1^2 + x_2^2 \leq 1 \wedge x_1 \geq 0 \wedge x_1^2 + 2x_1x_2 + x_2^2 \geq 2)$$

QE で x_2 を消去すると,

$$2x_1^2 = 1 \wedge x_1 \geq 0$$

が得られる. したがって, $x_1 = 1/\sqrt{2}$ が得られる. x_2 も同様に計算できる.

例 2

以下のパラメトリック最適化問題を考える.

$$\begin{array}{ll} \text{minimize} & -x_1 - \theta \\ \text{subject to} & x_1 \geq 0 \ \wedge \ \theta \geq 0 \ \wedge \ x_1^2 + \theta^2 \leq 1 \end{array}$$

目的関数の実行可能領域は以下のように定式化される.

$$\exists x_1 \ (y = -x_1 - \theta \ \wedge \ x_1 \geq 0 \ \wedge \ \theta \geq 0 \ \wedge \ x_1^2 + \theta^2 \leq 1)$$

QE を用いて決定変数 x_1 を消去すると, 以下のような論理式を得る.

$$\psi_{Feasible}(\theta, y) \equiv y^2 + 2\theta y + 2\theta^2 \leq 1 \ \wedge \ y \leq \theta \wedge \ 0 \leq \theta \leq 1$$

図 4 は $\psi_{Feasible}$ を描画したものである. 目的関数の最小値のパラメータ表現は以下のように定式化できる.

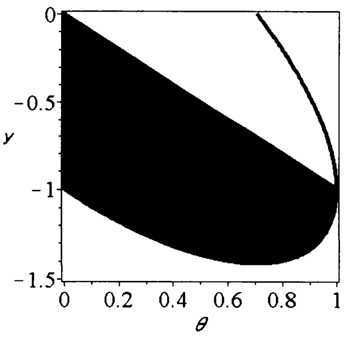

図 4: 例 2 の目的関数の実行可能領域

$$\begin{aligned} & \psi_{Feasible}(\theta, y) \wedge \neg\exists z\ (\psi_{Feasible}(\theta, z)\ \wedge\ z < y) \\ = \ & \forall z(\psi_{Feasible}(\theta, y) \wedge (\psi_{Feasible}(\theta, z)\ \rightarrow\ (z \geq y))) \\ = \ & \forall z(y^2 + 2\theta y + 2\theta^2 \leq 1 \wedge y \leq \theta \wedge 0 \leq \theta \leq 1 \wedge ((z^2 + 2\theta z + 2\theta^2 \leq 1 \wedge z \leq \theta) \rightarrow (z \geq y))) \end{aligned}$$

変数 z を消去すると以下の論理式が得られる．

$$\psi_{Pareto}(\theta, y) \equiv (y^2 + 2\theta y + 2\theta^2 = 1 \wedge y \leq \theta \wedge 0 \leq \theta \leq 1)$$

最小値を与える最適解のパラメータ表現は以下のように定式化される．

$$\exists y\ (y = -x_1 - \theta\ \wedge\ x_1 \geq 0\ \wedge\ \theta \geq 0\ \wedge\ x_1^2 + \theta^2 \leq 1\ \wedge\ \psi_{Pareto}(\theta, y))$$

QE を適用して変数 y を消去すると, 最適解を表す論理式

$$x^2 + \theta^2 = 1\ \wedge\ x \geq 0$$

を得ることができる．

例 3
次の多目的最適化問題を考える．

$$\begin{array}{ll} \text{minimize} & (x_1^2 + x_2^2, 5 + x_2^2 - x_1) \\ \text{subjectto} & -5 \leq x_1 \leq 5\ \wedge\ -5 \leq x_2 \leq 5 \end{array}$$

目的関数の可能領域は一解述語論理式で記述される．

$$\exists x_1 \exists x_2\ (y_1 = x_1^2 + x_2^2 \wedge y_2 = 5 + x_2^2 - x_1 \wedge -5 \leq x_1 \leq 5 \wedge -5 \leq x_2 \leq 5)$$

決定変数 x_1,x_2 を消去すると, 以下の論理式 $\psi_{Feasible}(y_1, y_2)$ を得る．

$$\begin{aligned} & -y_2 + y_1 - 25 \leq 0 \wedge 4y_2 - 4y_1 - 21 \leq 0 \wedge 0 \leq y_1 \leq 50 \wedge (\\ & \quad (-y_2 + 5 \leq 0 \wedge -4y_1 + 1 \leq 0 \wedge 4y_1 - 101 \leq 0) \vee \\ & \quad (y_2^2 - 10y_2 - y_1 + 25 \leq 0) \vee \\ & \quad (-y_2^2 + 60y_2 + y_1 - 925 \leq 0 \wedge y_2 - 30 \leq 0 \wedge -4y_1 + 101 \leq 0) \vee \\ & \quad (-y_2 + y_1 - 15 \leq 0 \wedge y_2^2 - 60y_2 - y_1 + 925 \leq 0)) \end{aligned}$$

図 5 は目的関数の実行可能領域 $\psi_{Feasible}$ を描画したものである．

パレート・フロントを表す一解述語論理式は $\psi_{Feasible}$ を用いて, 以下のように表される．

$$\forall z_1 \forall z_2\ (\psi_{Feasible}(y_1, y_2) \wedge (\neg\psi_{Feasible}(z_1, z_2) \vee (z_1 > y_1 \vee z_2 > y_2 \vee (z_1 \geq y_1 \wedge z_2 \geq y_2))))$$

QE を適用して z_1, z_2 を消去すると以下の論理式が得られる．

$$y_2^2 - 10y_2 - y_1 + 25 = 0 \wedge 0 \leq y_1 \leq 25 \wedge y_2 \leq 5$$

最後にパレート最適解を求める．パレート最適解は以下のように定式化される．

$$\begin{aligned} & \exists y_1 \exists y_2((y_1 = x_1^2 + x_2^2 \wedge y_2 = 5 + x_2^2 - x_1)\ \wedge\ (-5 \leq x_1 \leq 5 \wedge -5 \leq x_2 \leq 5) \wedge \\ & \quad (y_2^2 - 10y_2 - y_1 + 25 = 0 \wedge 0 \leq y_1 \leq 25 \wedge y_2 \leq 5)) \end{aligned}$$

変数 y_1, y_2 を QE で消去することで, 以下のパレート最適解を表す論理式が得られる．

$$0 \leq x_1 \leq 5 \wedge x_2 = 0$$

制約条件を満たす点集合はコンパクトであると仮定したが, 最小値が存在しない場合に (3) に QE を適用すると偽 (false) を復帰するため, 注意が必要である．最小値を持たない場合にも極小値を定式化することは可能であるが本稿では説明は省略する．

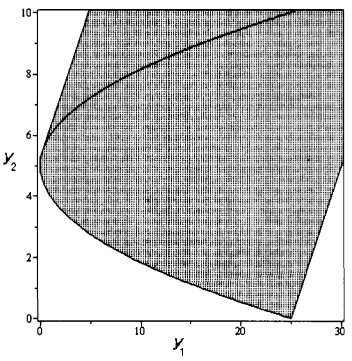

図 5: 例 3 の目的関数の実行可能領域

3.1 QE で扱える最適化問題

QE は一階述語論理式を扱うアルゴリズムであるため, 多項式最適化問題よりも広いクラスの問題を扱うことができる. 以下にその例を紹介する. ここでは単目的な場合を用いて紹介しているが, 多項式最適化問題に対する場合と同様に多目的およびパラメトリックの場合にも適用可能である.

3.1.1 分数最適化問題

以下のような分数最適化問題を考える.

$$\begin{array}{ll} \text{minimize} & f(\boldsymbol{x})/g(\boldsymbol{x}) \\ \text{subject to} & \varphi(\boldsymbol{x}) \end{array}$$

ここで $f(\boldsymbol{x})$, $g(\boldsymbol{x})$ は多項式である.

多項式最適化問題のときと同様に, 新しい変数 y を用いて y が目的関数の実行可能領域に属する場合に真となるような以下の一階述語論理式を考える.

$$\exists \boldsymbol{x}\ (g(\boldsymbol{x})y = f(\boldsymbol{x}) \wedge g(\boldsymbol{x}) \neq 0 \wedge \varphi(\boldsymbol{x}))$$

この論理式から決定変数 $\boldsymbol{x}$ を消去することで y に関する実行可能領域を正確に求めることができる.

3.1.2 ミニマックス最適化問題

QE を用いると以下のようなミニマックス最適化問題も正確に解くことができる.

$$\begin{array}{ll} \text{minimize} & \max(f_1(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x})) \\ \text{subject to} & \varphi(\boldsymbol{x}) \end{array}$$

ミニマックス最適化問題を扱う場合には以下の一階述語論理式を考える.

$$\exists \boldsymbol{x}\ (y \geq f_1(\boldsymbol{x}) \wedge \cdots \wedge y \geq f_m(\boldsymbol{x}) \wedge \varphi(\boldsymbol{x}))$$

この場合は新しい変数 y は目的関数の実行可能領域を表していないが, y の実行可能領域の大域的最小値は与えられた最適化問題の大域的最小値と一致する. したがって, QE を用いて変数 $\boldsymbol{x}$ を消去することで最小値を正確に得ることができる.

4 おわりに

数式処理による多項式最適化問題の解法について紹介した．QE は論理式の等価変換を行うため，数値手法が得意でない非凸な問題でも正確に解くことが可能である．また変数とパラメータを区別しないため，パラメトリック最適化問題，多目的最適化問題も同様に正確に解くことができる．さらに，QE が扱うのは一階述語論理式であるため分数最適化問題のような多項式最適化問題よりも広いクラスの問題を解くことも可能である．

QE の唯一の問題はその計算量にある．しかし，アルゴリズムの改良と計算機の進歩により QE で解ける範囲は着実に広がっている．また，特定の誤差を許した手法や数値手法と組み合わせた手法などが提案されている [11, 12, 13, 15].

最後に，MIP ソルバなどの数値手法による数理最適化ソルバと同様に QE アルゴリズムのクセを知っているほうがより大きな問題を解けることがある．そのため QE ツールで問題が解けないがある場合にはお知らせいただけると幸いである．

参考文献

[1] H. Anai and S. Hara. Fixed-structure robust controller synthesis based on sign definite condition by a special quantifier elimination. In *Proceedings of American Control Conference, 2000*, volume 2, pages 1312–1316, 2000.

[2] Hirokazu Anai and Kazuhiro Yokoyama. Cylindrical algebraic decomposition via numerical computation with validated symbolic reconstruction. In Andreas Dolzman, Andreas Seidl, and Thomas Sturm, editors, *Algorithmic Algebra and Logic*, pages 25–30, 2005.

[3] Christopher W. Brown. *Solution formula construction for truth invariant cad's.* PhD thesis, University of Delaware Newark, 1999.

[4] Christopher W. Brown. QEPCAD B: A program for computing with semi-algebraic sets using CADs. *SIGSAM BULLETIN*, 37:97–108, 2003.

[5] Bob F. Caviness and Jeremy R Johnson, editors. *Quantifier elimination and cylindrical algebraic decomposition.* Texts and Monographs in Symbolic Computation. Springer, 1998.

[6] George E. Collins. *Quantifier elimination and cylindrical algebraic decomposition*, pages 8–23. In Caviness and Johnson [5], 1998.

[7] George E. Collins and Hoon Hong. Partial cylindrical algebraic decomposition for quantifier elimination. *Journal of Symbolic Computation*, 12(3):299–328, 1991.

[8] George E. Collins, Jeremy R. Johnson, and Werner Krandick. Interval arithmetic in cylindrical algebraic decomposition. *Journal of Symbolic Computation*, 34(2):145–157, 2002.

[9] James H. Davenport and Joos Heintz. Real quantifier elimination is doubly exponential. *Journal of Symbolic Computation*, 5(1/2):29–35, 1988.

[10] Andreas Dolzmann and Thomas Sturm. Redlog: computer algebra meets computer logic. *SIGSAM Bull.*, 31:2–9, June 1997.

[11] Jean-Charles Faugère, Guillaume Moroz, Fabrice Rouillier, and Mohab Safey El Din. Classification of the perspective-three-point problem, discriminant variety and real solving polynomial systems of

inequalities. In *Proceedings of the twenty-first international symposium on Symbolic and algebraic computation*, ISSAC '08, pages 79–86, New York, NY, USA, 2008. ACM.

[12] H. Hong and M. Safey El Din. Variant quantifier elimination. *Journal of Symbolic Computation*, pages 1–24, 2011. to appear.

[13] Hidenao Iwane, Akifumi Kira, and Hirokazu Anai. Construction of explicit optimal value functions by a symbolic-numeric cylindrical algebraic decomposition. In Vladimir P. Gerdt, Wolfram Koepf, Ernst W. Mayr, and Evgenii V. Vorozhtsov, editors, *CASC*, volume 6885 of *Lecture Notes in Computer Science*, pages 239–250. Springer, 2011.

[14] Hidenao Iwane, Hitoshi Yanami, and Hirokazu Anai. An effective implementation of a symbolic-numeric cylindrical algebraic decomposition for optimization problems. In *Proceedings of the 2011 International Workshop on Symbolic-Numeric Computation*, volume 1, pages 168–177, 2011.

[15] Hidenao Iwane, Hitoshi Yanami, and Hirokazu Anai. A symbolic-numeric approach to multi-objective optimization in manufacturing design. *Mathematics in Computer Science*, 2011. in press.

[16] Hidenao Iwane, Hitoshi Yanami, Hirokazu Anai, and Kazuhiro Yokoyama. An effective implementation of a symbolic-numeric cylindrical algebraic decomposition for quantifier elimination. In *Proceedings of the 2009 International Workshop on Symbolic-Numeric Computation*, volume 1, pages 55–64, 2009.

[17] Rüdiger Loos and Volker Weispfenning. Applying linear quantifier elimination. *The Computer Journal*, 36(5):450–462, 1993.

[18] Scott McCallum. An improved projection operator for cylindrical algebraic decomposition. In Caviness and Johnson [5], pages 242–268.

[19] Scott McCallum. On propagation of equational constraints in CAD-based quantifier elimination. In *Proceedings of the 2001 international symposium on Symbolic and algebraic computation*, ISSAC '01, pages 223–231, New York, NY, USA, 2001. ACM.

[20] Adam W. Strzeboński. Cylindrical algebraic decomposition using validated numerics. *Journal of Symbolic Computation*, 41(9):1021–1038, 2006.

[21] Alfred Tarski. *A decision method for elementary algebra and geometry*. University of California Press, 2nd edition, 1952.

[22] Hitoshi Yanami and Hirokazu Anai. The maple package SyNRAC and its application to robust control design. *Future Generation Computer Systems*, 23(5):721–726, 2007.

[23] 穴井宏和 and 横山和弘. 計算実代数幾何入門. In **数学セミナー**, pages 64–70. 日本評論社, 11 2007.

[24] 穴井 宏和 and 横山 和弘. *QE* **の計算アルゴリズムとその応用 – 数式処理による最適化**. 東京大学出版会, 8 2011.

[25] 野呂正行. **計算機代数入門**. Number 9 in Rokko lectures in mathematics. 神戸大学理学部数学教室, 2000.